09/29/00 JC693 U.S. PTO

10-02-00 A.

PATENT APPLICATION TRANSMITTAL LETTER
(Large Entity)

Docket No.
P24324 USA

09/29/00 JC925 U.S. PTO 09/675790

TO THE ASSISTANT COMMISSIONER FOR PATENTS

Transmitted herewith for filing under 35 U.S.C. 111 and 37 C.F.R. 1.53 is the patent application of:

DONALD F. FERGUSON, ANAND BYRAPPAGARI AND GRAEME DIXON

For: **SYSTEM AND METHOD FOR UPGRADING SOFTWARE IN A DISTRIBUTED COMPUTER SYSTEM**

Enclosed are:

- ☒ Certificate of Mailing with Express Mail Mailing Label No. **EL592133389US**
- ☒ **Four (4)(informal)** sheets of drawings.
- ☐ A certified copy of a application.
- ☒ Declaration ☒ Signed. ☐ Unsigned.
- ☒ Power of Attorney
- ☒ Information Disclosure Statement
- ☐ Preliminary Amendment
- ☒ Other: **Assignment, PTO 1449 (with 7 references) and acknowledgment postcard**

CLAIMS AS FILED

For	#Filed	#Allowed	#Extra		Rate	Fee
Total Claims	20	- 20 =	0	x	$18.00	$0.00
Indep. Claims	4	- 3 =	1	x	$78.00	$78.00
Multiple Dependent Claims (check if applicable) ☐						$0.00
					BASIC FEE	$690.00
					TOTAL FILING FEE	$768.00

☐ A check in the amount of to cover the filing fee is enclosed.

☒ The Commissioner is hereby authorized to charge and credit Deposit Account No. **09-0461** as described below. A duplicate copy of this sheet is enclosed.

- ☒ Charge the amount of **$768.00** as filing fee.
- ☒ Credit any overpayment.
- ☒ Charge any additional filing fees required under 37 C.F.R. 1.16 and 1.17.
- ☐ Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).

Dated: **September 29, 2000**

Gregory S. Bernabeo
Signature

Gregory S. Bernabeo, Esquire
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107-2950
Telephone: 215-923-4466
Facsimile: 215-923-2189

cc:

JC925 U.S. PTO
09/675790
09/29/00

CERTIFICATE OF MAILING BY EXPRESS MAIL UNDER 37 CFR 1.10

EL592133389US
Express Mail Label Number

S&L Docket No. P24234 USA
(RSW9-2000-0119-US1)

I hereby certify that the attached Patent Application Transmittal Letter for Large Entity Concerning Filing Under 35 U.S.C. 111 and 37 C.F.R. 1.53, including a duplicate copy of the Transmittal Letter, Patent Application specification including 4 sheets of informal drawings, an Abstract, an Information Disclosure Statement, PTO 1449 with 7 references, an Assignment, a Declaration and Power of Attorney for Patent Application, and an acknowledgment postcard are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" under 37 C.F.R. § 1.10 in an envelope addressed to the Box Patent Application, Commissioner for Patents, Washington, DC 20231 on September 29, 2000.

Ethan D. Tripp
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107-2950
Telephone: (215) 923-4466
Facsimile: (215) 923-2189

M:\GBernabeo\Ibm\P24324.usa\PatentOffice\ExpressMailCertificate.wpd

Express Mail Label No. EL592133389US

Patent RSW9-2000-0119-US1

SYSTEM AND METHOD FOR UPGRADING SOFTWARE IN A DISTRIBUTED COMPUTER SYSTEM

FIELD OF THE INVENTION

The present invention relates generally to the field of client/server (also known as "distributed") computer systems. In particular, the present invention relates to a system and method for upgrading software, e.g., on a server, in a distributed computer system.

BACKGROUND OF THE INVENTION

The client/server computing model is a distributed computing model allowing a client process running on a client computer (a "client"), to delegate some of its work to a server process running on a server computer (a "server"). Distributed computer systems typically use multiple servers to provide essential functions to multiple users' clients in multiple concurrent user sessions. The use of multiple servers improves server availability and system capacity. It is often essential that distributed computer systems provide service on a continuously available, uninterrupted basis. However, distributed computer systems require periodic maintenance, including software upgrades.

Figure 1 illustrates an exemplary distributed communications system 10 of the prior art. The system 10 includes three clients 12, 14, 16 connected to a communications network 18, e.g. the Internet or an intranet. Also connected to the communications network 18 is a

server cluster 20 including application servers 22, 24 and 26. Each application server in the server cluster 20 stores an identical or nearly identical set of computer software programs and components thereof (collectively, "software"). Such software either enables a server to perform processes and/or interact with clients or is application software stored to be served to clients. An older version of such software is periodically replaced with a newer version providing new or more reliable functionality. This "upgraded" software must typically be installed on multiple servers in the distributed system.

In operation, individual clients use software applications stored on the application servers by sending data packets among one another via the communications network. Clients send messages, or "requests" for the software and receive response messages from the servers. Individual requests are part of a "session." Application software starts and ends sessions. Since individual client's requests within a session may be routed to different application servers in the server cluster, some mechanism for sharing the session information is necessary. Various methods, both hardware and software based, are well known for sharing such information, including allowing access to shared memory or databases, and broadcasting session updates to all application servers in the cluster. For example, such information is typically stored in shared storage on an external data server 28.

Each application server maintains a list of active client sessions (the "Active Session List") that it is currently supporting. Effectively, each entry in the Active Session List stores a unique session identification code ("session ID") and associated session data for each client session. When an application server detects a new client session, it writes a new entry to the Active Session List. At the end of a request, associated session data is written to the client.

Additionally, each application server stores information reflecting an active session count reflecting the number of active sessions being supported.

Preferably, such Active Session Lists from each application server are compiled into a Session List maintained in a shared storage area, such as at an external data server 28 for returning data to the clients, as is well known in the art. When a next request from the client is to be serviced by a different application server, that application server reads the session ID from the Session List. The basic execution model is: receive client request, load session data from the Session List, process request, return response, write session to Session List.

The application servers in the server cluster 20 are connected to the communications network 18 via a router 30. The router 30 and/or network routing layer functions in a well known manner, operating to distribute clients' requests among the multiple application servers.

A system management server 32 is connected to the router 30 and each of the applications servers. The system management server stores a mapping of applications/software programs to application servers (the "Server List"). This mapping information is accessed by the router to complete routing functions. The system management server provides configuration and health/load information to the router of the communications network 18.

There are several known architectures for distributed computer systems. One known example is the Java 2 Enterprise Edition specification of a client/server architecture (Java 2 Enterprise Edition™ is a trademark of Sun Microsystems, Inc.). The Java 2 Enterprise Edition™ specification allows server vendors to support seamless migration of all ongoing client sessions from a first server to a second server to permit the first server to be removed

from the distributed computer system or by-passing an unscheduled server failure without aborting a pending client session. Figure 2 shows an exemplary flow diagram 50 depicting the process of upgrading software in the distributing computing system of Figure 1.

As shown in Figure 2, the upgrade process begins with maintenance of the Session List and Server List, as shown at steps 51 and 52. There are many well known techniques for maintaining the Session List and Server List. Next, at step 54, an application server storing software to be upgraded is identified, e.g., Application Server 1. For example, the application server may be identified by reference to the Server List maintained in step 52.

The system management server 32 then forbids the router to route client requests to Application Server 1, as shown at step 56, thereby removing Application Server 1 from active service during the upgrade procedure. For example, the system management server 32 may send a signal to the router 30 with an appropriate message to cause the router 30 to stop sending new requests which could be serviced by the application server being upgraded. This may require an addition, deletion or modification to the Server List stored at the router and/or the system management server.

The system management server 32 then monitors whether the application server is currently supporting any active client requests, as shown at step 58. For example, the system management server may send an appropriate signal to the application server to determine if the active session count is 0. If the application server is still supporting active client requests (i.e., the active session count is greater than 0), the system management server waits or continues polling, as shown at step 60. This continues until the active session count drops to 0, indicating that the application server is no longer supporting any active client requests. Since the router will send no new requests to Application Server 1 and the current sessions

have ended, Application Server 1 is effectively removed from the distributed computing system.

At this point, Application Server 1's software may be upgraded without disrupting any pending client requests. When the application server is no longer supporting any client requests, the appropriate software program on the appropriate application server is upgraded, as shown in step 62. Methods and computer hardware and software for upgrading software are well known in the art.

Once the upgrade is complete, the system management server 32 then permits the router 30 to route new client requests to the Application Server 1, as shown at step 64. This may be achieved by addition, deletion or modification of the Server List. For example, step 64 may be performed responsive to a signal sent from Application Server 1 to the system management server to indicate that the upgrade is complete. Finally, the method ends, as shown at step 65. In effect, the application server is added back to the distributed computing system.

Accordingly, prior art methods remove all of an application server's resources from a distributed computing system in order to update a single software application, which may constitute a small fraction of the application server's stored applications. Accordingly, the distributed computer system's burden is increased during a software upgrade process because the system must service client's requests with one fewer application server.

SUMMARY OF THE INVENTION

The present invention provides a system and method for upgrading software in a distributed computer system. Specifically, the present invention provides a system and

method for selectively redirecting or "migrating" only client sessions for the to-be-upgraded software. This is achieved by modifying software at the application server(s) and the system management server to cause the Server List to track servers capable of handling requests on the basis of each individual piece of software, and to cause the Session List to track requests on the basis of each individual piece of software. Accordingly, requests for only a specific piece of software can be migrated when only a small portion of an application server's resources need to be upgraded, the remainder of the application server's resources remaining available to the distributed computing system. In other words, the application server on which the software being upgraded resides can continue servicing requests for other pieces of software. As compared with prior art methods, the present invention reduces the burden on the distributed computing system during a software upgrade process.

In accordance with the method, the application server is prevented from receiving any new requests for a particular computer program that needs to be upgraded. This is achieved by maintaining a mapping of computer programs to application servers and for a given server, instructing a router not to send requests for the particular computer program to the given server. The system then waits until all current requests serviced by the application server for that computer program have ended. This is achieved by tracking active client requests on the basis of each computer program; for the particular computer program, it is determined whether any client requests are currently active. Then, after acknowledging completion of upgrading of the computer program, the application server is permitted to receive any new requests for the computer program. For example, the application server may inform the system management server that the upgrade is complete, at which time the system

management server instructs the router that it may resume routing client requests for the particular computer program to the upgraded application server.

In one embodiment of the present invention, the system management server includes software for systematically upgrading a plurality of servers so that the upgrade process "ripples" across the distributed computing system by refusing requests for the to-be-upgraded software sequentially for multiple servers.

Apparatuses for carrying out the inventive method are provided.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 a block diagram of a distributed computing system in accordance with the prior art;

Figure 2 is an exemplary flow diagram of a software upgrade method in accordance with the prior art;

Figure 3 is an exemplary flow diagram of a software upgrade method in accordance with the present invention;

Figure 4 is a block diagram of an application server in accordance with the present invention; and

Figure 5 is a block diagram of an application server in accordance with the present invention.

DETAILED DESCRIPTION

The present invention provides a system and method for upgrading software in a distributed computer system including an application server storing to-be-upgraded software.

Prior art techniques tracked application server usage on a session basis and required an entire application server to be "quiesced", i.e., placed on an inactive status such that no new clients' requests for software stored by that application server could be routed to that application server during the upgrade process. Accordingly, the quiesced application server could not service any clients' requests. The present invention provides a system and method for permitting limited quiescence of an application server by tracking application server usage on a per application basis. Accordingly, an application server storing several software programs is permitted to continue servicing clients requests for software programs other than the to-be-upgraded software program while the to-be-upgraded software is upgraded. Only clients' requests for the to-be-upgraded software are routed away from the application server during the upgrade process. Accordingly, when only one software program stored on the application server needs to be upgraded, the remainder of the software programs stored on the application server remain available to the distributed computing system. As compared with prior art methods, the present invention therefore reduces the burden on the distributed computing system during a software upgrade process.

Figure 3 is an exemplary flow diagram 70 of a software upgrade method in accordance with the present invention. As shown in Figure 3, the method begins with maintenance of a Session List and Server List, as shown at steps 71 and 72. This step differs from the prior art in that each entry in the Session List includes identification of the software program associated with the request, i.e., a mapping of each active request to each application. In one embodiment, the Session List is maintained only at each application server. In another embodiment, each application server's request list compiled into a master Session List at the external data server or in other shared storage. Any embodiment in which

it is apparent when the application server is no longer servicing a client's request for the computer program is suitable.

Additionally, each entry in the Server List includes identification of the software programs on each application server, i.e., a mapping of each application to each application server. The Server List is preferably stored at the system management server and is used to provide the router 30 with a mapping of applications to application servers. This requires modification of software at the application servers and the system management server to track request/session information on a per application basis. For example, in the Internet context, the Session List may include a mapping of URL's for each software program to each server. Such mapping information is also stored in the Server List stored by the system management server. Alternatively, each software program may be identified by filename, transaction code, etc.

An application server storing the to-be-upgraded ("TBU") software is then identified, as shown at step 74. This may be accomplished with or without human intervention, e.g., by observing which software programs are stored on a computer and checking for the to-be-upgraded software. Alternatively, an application server may be identified in an automated fashion by maintaining a database of software programs stored on each application server, the database being referenced by a software program for systematically checking and upgrading each application server in a server cluster, etc. For example, such a software program may be stored and executed at the system management server.

Conceptually, once the TBU software has been identified at step 74, the router 30 is then forbidden to rout new client requests for the TBU software program to the to-be-upgraded application server, as shown at step 76. This step may be implemented, for

example, by sending an appropriate signal to the router 30 to indicate that requests for certain URL's cannot be routed to the TBU application server and/or by making an appropriate entry in the Server List stored at the system management server and/or router to indicate that the TBU application server is no longer accepting requests for the TBU software. Accordingly, when a new client's request for the TBU software is sent to the system management server, the Server List is consulted, and the request will be routed to an application server in the server cluster other than the TBU application server.

The next step is to determine whether the TBU application server is currently supporting a client request for the software, as shown at step 78. This needs to be determined because the TBU software cannot be upgraded if the TBU application server is still supporting active client requests for the TBU software. This can be determined by reference to the Session List, which tracks requests on a per application basis. If the TBU application server is currently servicing a request for the software, the system waits, as shown at step 80. When the TBU application server is no longer servicing a request for the TBU software, the TBU software program is no longer in use and is ready to be upgraded.

Accordingly, the software is upgraded on the TBU application server, as shown at step 82. This may be accomplished in any suitable manner, as is well-known in the art.

Finally, the router 30 is permitted to route new client requests for the software to the application server storing the now-upgraded software, as shown at step 34, and the process ends, as shown at step 35. This may be accomplished by modifying the Server List stored at the system management server and/or router to reflect that the application server is no longer inactive or quiesced for client requests for the upgraded software.

In this manner, clients' requests are continuously serviced, without interruption,

during the upgrade process, and with an overall reduced burden on the distributed computing system, as compared with prior art methods.

In a preferred embodiment, these steps are carried out in an automated and systematic fashion by execution of a suitable computer software program such that upon completion of the upgrading of the software on a first application server, the method repeats for a second application server storing the to-be-upgraded software program. In this manner, the upgrade "ripples" across the server cluster.

It should be understood that the present invention can also be used to upgrade not only the application software stored by an application server but also the server software itself. In such an embodiment, application software code is analyzed to determine a Components list and information is stored in the Server List and Session List to reflect which components of the server software is or will be used by to service the client's requests. The TBU component is then treated like application software for the purpose of the upgrade process in accordance with the present invention. When the server software is stored on an application server, the components list is generated to reflect the relationship of the components and how they will be used to service clients' requests for software. The Components List is preferably stored on the system management server.

Figure 4 is a block diagram of an application server in accordance with the present invention. The hardware of the application server is of a type generally known in the art. The application server 100 includes a central processing unit ("CPU") 102, a memory 104, e.g., random access memory ("RAM"), read only memory ("ROM") and/or a storage device such as a hard disk drive, and a telecommunications device 106 for communicating via a communications network, e.g., using TCP/IP technology. The telecommunications device

may include a modem and/or a network card connected via a communications port 108. The application server may optionally include video display devices and/or input devices.

The application server 100 stores in its memory a first program for storing in said memory a list of active client requests supported by said application server, a second program for storing in said memory an identification of a computer program associated with each active client request, a third program for identifying a computer program upon receipt of a signal via said telecommunications device, and a fourth program for sending a signal via said telecommunications device when said application server is no longer supporting an active client request for said computer program.

Figure 5 is a block diagram of a system management server 150 in accordance with the present invention. The hardware of the system management server 150 is of the type generally known in the art. The system management server 150 includes a central processing unit ("CPU") 152, a memory 154, e.g., random access memory ("RAM"), read only memory ("ROM") and/or a storage device such as a hard disk drive, and a telecommunications device 156 for communicating via a communications network, e.g., using TCP/IP technology. The telecommunications device may include a modem and/or a network card connected via a communications port 158. The system management server 150 may optionally include video display devices and/or input devices.

The system management server 150 stores in its memory a first program for storing in said memory a list of application servers for servicing requests for a computer program, a second program for storing in said memory an identification of a computer program associated with each active client request, a third program for sending a signal via said telecommunications device to instruct a router to stop routing client requests for a particular

computer program to a particular application server, and a fourth program for sending a signal via said telecommunications device to instruct a router to begin routing client requests for said particular computer program to said particular application server responsive to an indication that an upgrade process of said particular computer program on said particular application server is complete.

Having thus described a few particular embodiments of the invention, various alterations, modifications, and improvements will readily occur to those skilled in the art. Such alterations, modifications and improvements as are made obvious by this disclosure are intended to be part of this description though not expressly stated herein, and are intended to be within the spirit and scope of the invention. Accordingly, the foregoing description is by way of example only, and not limiting. The invention is limited only as defined in the following claims and equivalents thereto.

What is claimed is:

1. A method for upgrading at least one of a plurality of computer programs stored on an application server in a distributed computing environment, said method comprising:

preventing said application server from servicing requests for an upgrade-ready computer program while permitting said application server to service client requests for other computer programs.

2. The method of claim 1, further comprising the steps of:

(a) preventing said application server from receiving any new requests for said computer program;

(b) waiting until all of said application server's current requests for said computer program have ended;

(c) acknowledging completion of upgrading of said computer program; and

(d) permitting said application server to receive any new requests for said computer program.

3. The method of claim 2, wherein step (a) comprises the step of:

(e) sending a signal to a router to instruct said router to stop routing requests for said computer program to said application server.

4. The method of claim 2, wherein said signal in step (e) comprises an

identification code identifying said computer program.

5. The method of claim 4, wherein said identification code comprises a universal resource locator.

6. The method of claim 4, wherein said identification code comprises a filename.

7. The method of claim 2, wherein step (d) comprises the step of:

(f) sending a signal to a router to instruct said router to begin routing requests for said computer program to said application server.

8. The method of claim 7, wherein said signal in step (f) includes an identification code identifying said computer program.

9. The method of claim 2, wherein step (e) comprises the step of:

(g) maintaining a server list, said server list identifying a computer program and an application server for satisfying a request for said computer program; and

(h) identifying said computer program and said application server to said router.

10. The method of claim 9, wherein said signal in step (f) comprises an identification code read from said server list.

11. The method of claim 2, further comprising the step of:

(i) maintaining a session list of active client requests serviced by said application server, said session list identifying a client request and a computer program requested by said client.

12. The method of claim 11, wherein step (b) comprises the step of:

(j) referencing said session list.

13. The method of claim 2, further comprising the step of:

(k) repeating steps (a) through (d) for a next application server storing said computer program.

14. The method of claim 2, wherein step (c) comprises the step of:

(l) receiving a signal from an application server upon completion of upgrading of said computer program at said application server.

15. An application server for facilitating upgrading in a distributed computing environment, the application server comprising:

a central processing unit;

a memory operatively connected to said central processing unit;

a telecommunications device operatively connected to said central processing unit and capable of communicating via a communications network;

a first program stored in said memory and executable by said central processing unit for storing in said memory a list of active client requests supported by said application

server;

a second program stored in said memory and executable by said central processing unit for storing in said memory an identification of a computer program associated with each active client request;

a third program stored in said memory and executable by said central processing unit for identifying a computer program upon receipt of a signal via said telecommunications device; and

a fourth program stored in said memory and executable by said central processing unit for sending a signal via said telecommunications device when said application server is no longer supporting an active client request for said computer program.

16. A system management server for facilitating upgrading in a distributed computing environment, the application server comprising:

a central processing unit;

a memory operatively connected to said central processing unit;

a telecommunications device operatively connected to said central processing unit and capable of communicating via a communications network;

a first program stored in said memory and executable by said central processing unit for storing in said memory a list of application servers for servicing requests for a computer program;

a second program stored in said memory and executable by said central processing unit for storing in said memory an identification of a computer program associated with each active client request;

a third program stored in said memory and executable by said central processing unit for sending a signal via said telecommunications device to instruct a router to stop routing client requests for a particular computer program to a particular application server; and

a fourth program stored in said memory and executable by said central processing unit for sending a signal via said telecommunications device to instruct a router to begin routing client requests for said particular computer program to said particular application server responsive to an indication that an upgrade process of said particular computer program on said particular application server is complete.

17. A method for upgrading one of a plurality of computer programs stored on an application server in a distributed computing environment, said method comprising the steps of:

(a) instructing a router to stop routing requests for said computer program to said application server;

(b) waiting until said application server is no longer supporting a current client request for said computer program;

(c) after completion of upgrading of said computer program, instructing said router to begin routing requests for said computer program to said application server.

18. The method of claim 17, wherein step (a) comprises sending a signal to said router identifying said computer program and said application server, said signal comprising information retrieved from a server list and identifying said computer program.

19. The method of claim 17, wherein step (b) comprises receiving a signal from said application server, said application server maintaining a list of active client requests, said list identifying said computer program, said application server sending said signal when said list reflects no active client requests for said computer program.

20. The method of claim 17, wherein step (c) comprises sending a signal to said router identifying said computer program and said application server, said signal comprising information retrieved from a server list and identifying said computer program.

SYSTEM AND METHOD FOR UPGRADING SOFTWARE IN A DISTRIBUTED COMPUTER SYSTEM

ABSTRACT OF THE INVENTION

A method for upgrading one of several computer programs stored on an application server in a distributed computing environment while permitting the application server to continue to service requests for other computer programs. In accordance with the method, the application server is prevented from receiving any new requests for the computer program. The system then waits until all current requests serviced by the application server for that computer program have ended. Then, after acknowledging completion of upgrading of the computer program, the application server is permitted to receive any new requests for the computer program. Apparatuses for carrying out the inventive method are provided.

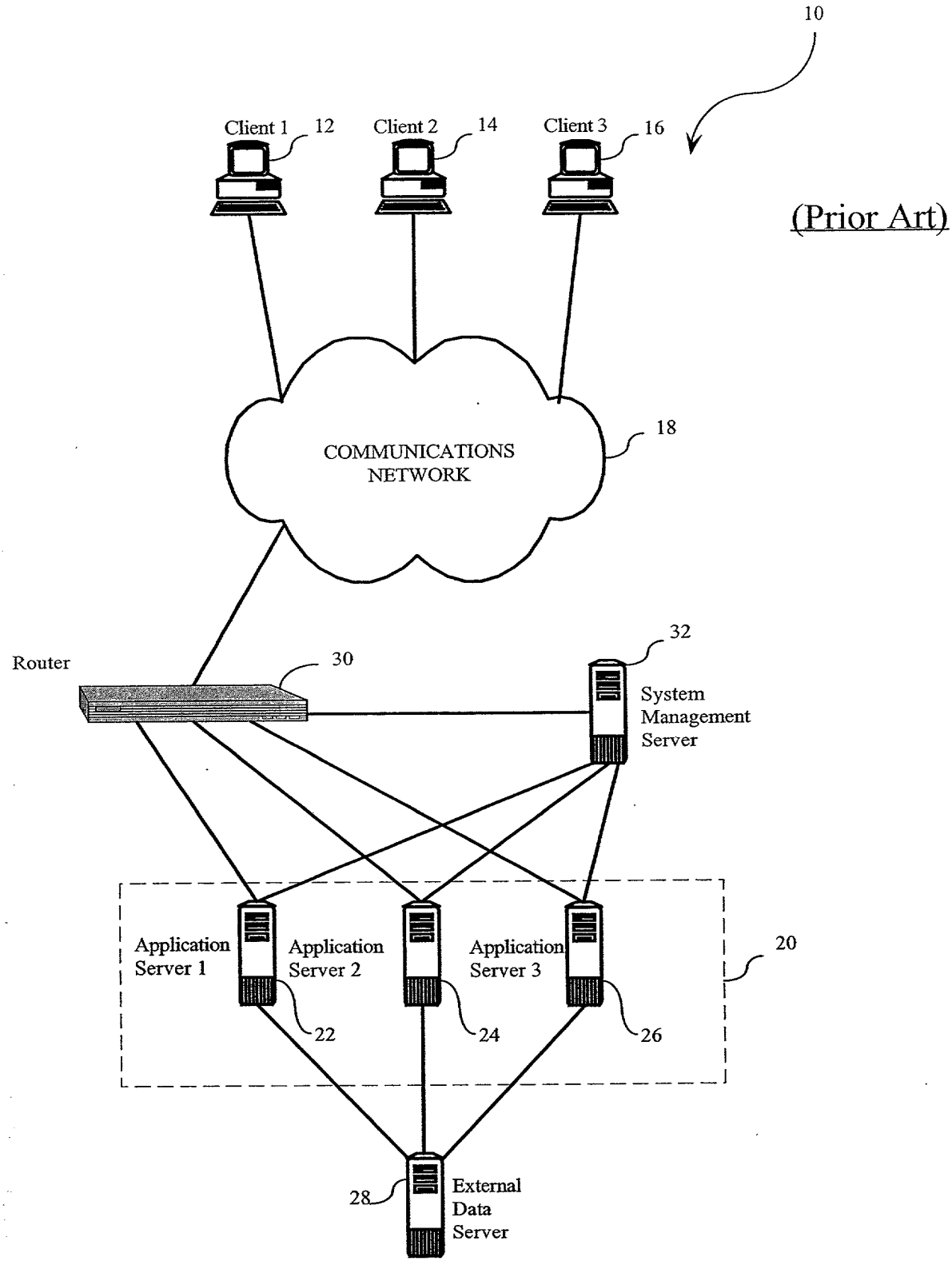
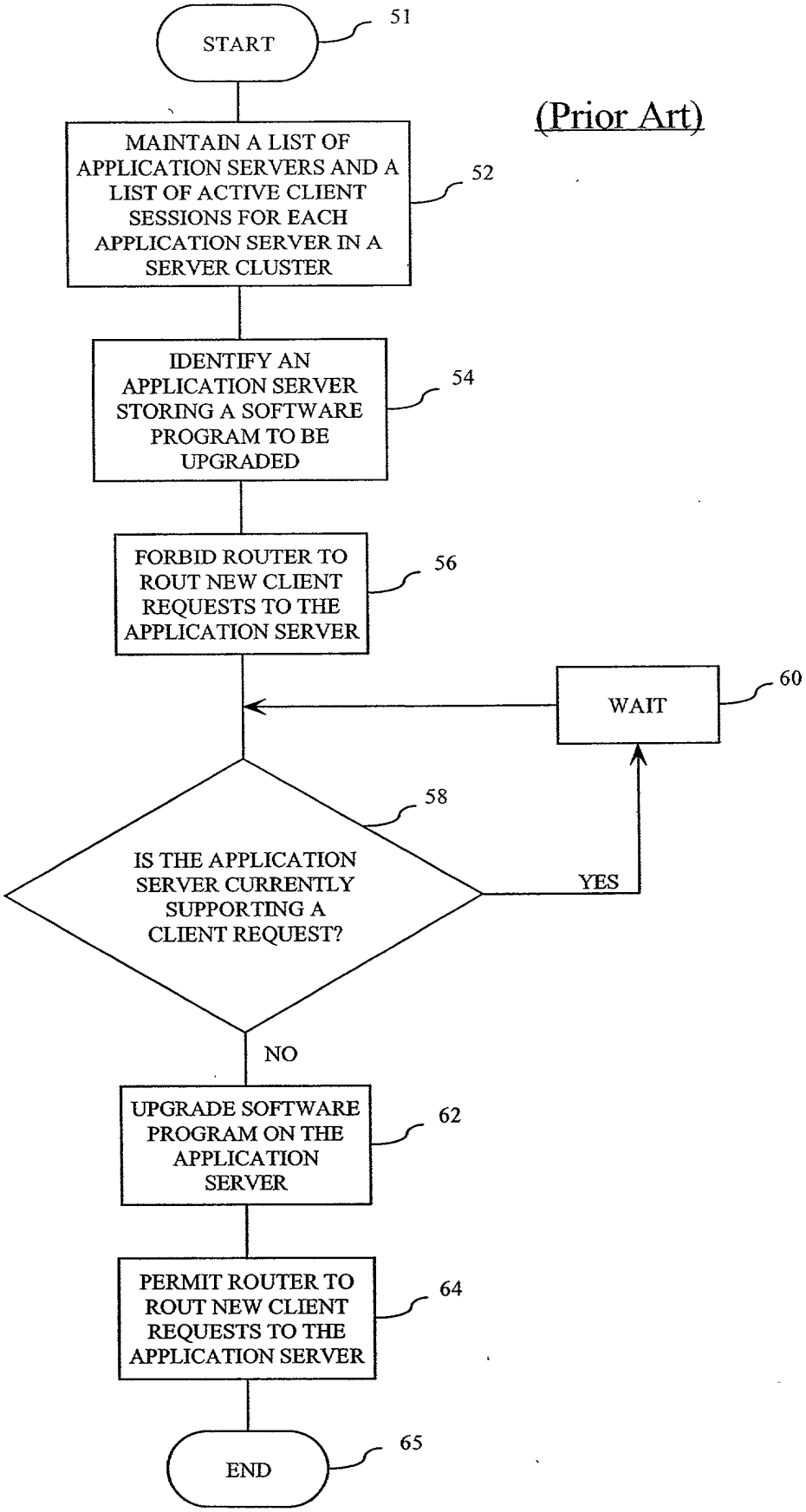

10
Client 1
12
Client 2
14
Client 3
16
(Prior Art)
COMMUNICATIONS NETWORK
18
Router
30
32
System Management Server
Application Server 1
Application Server 2
Application Server 3
20
22
24
26
28
External Data Server

Figure 1

PRIOR ART FLOW DIAGRAM 50



(Prior Art)

Figure 2

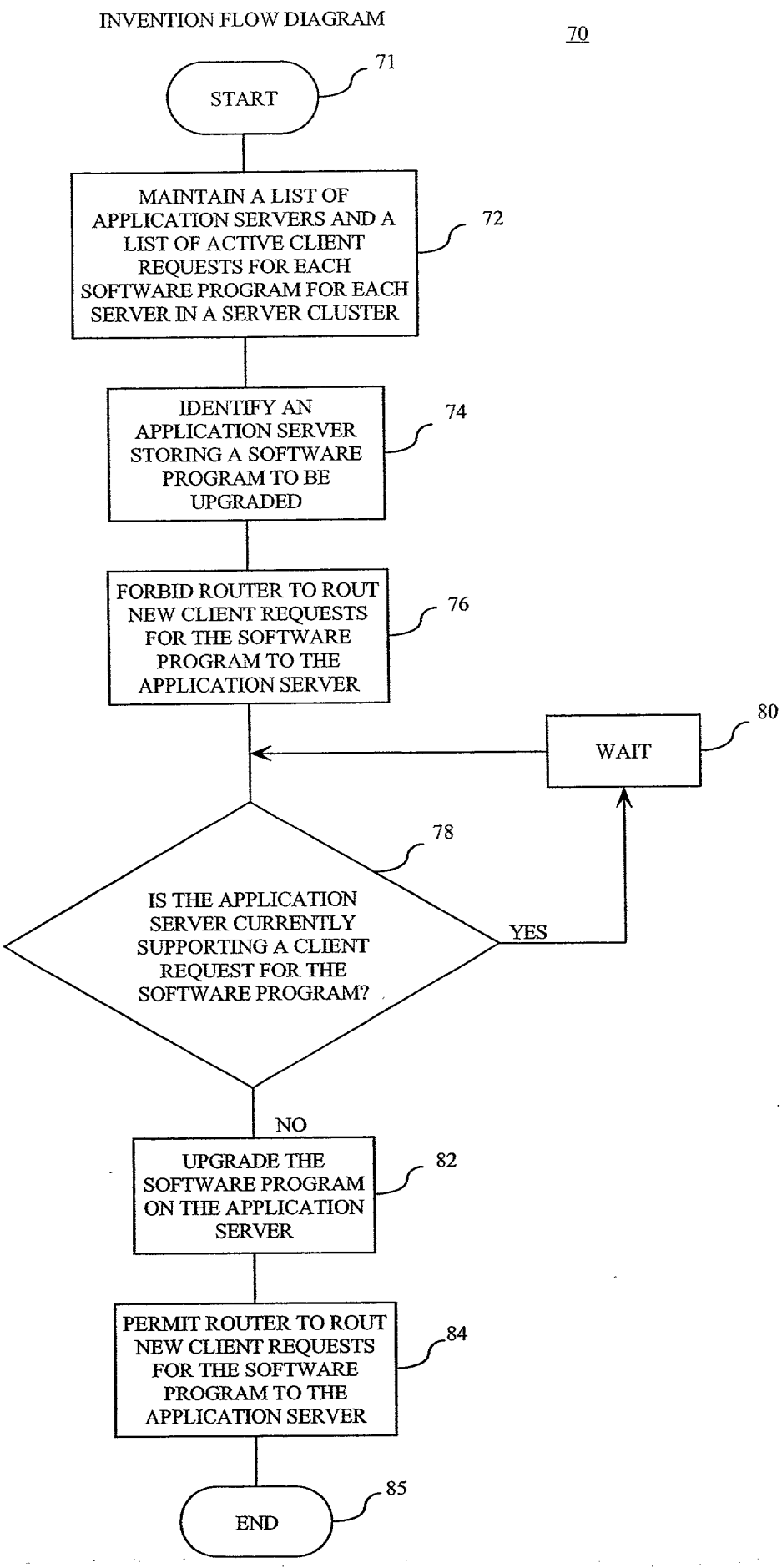

INVENTION FLOW DIAGRAM
70
START
71
MAINTAIN A LIST OF APPLICATION SERVERS AND A LIST OF ACTIVE CLIENT REQUESTS FOR EACH SOFTWARE PROGRAM FOR EACH SERVER IN A SERVER CLUSTER
72
IDENTIFY AN APPLICATION SERVER STORING A SOFTWARE PROGRAM TO BE UPGRADED
74
FORBID ROUTER TO ROUT NEW CLIENT REQUESTS FOR THE SOFTWARE PROGRAM TO THE APPLICATION SERVER
76
WAIT
80
IS THE APPLICATION SERVER CURRENTLY SUPPORTING A CLIENT REQUEST FOR THE SOFTWARE PROGRAM?
78
YES
NO
UPGRADE THE SOFTWARE PROGRAM ON THE APPLICATION SERVER
82
PERMIT ROUTER TO ROUT NEW CLIENT REQUESTS FOR THE SOFTWARE PROGRAM TO THE APPLICATION SERVER
84
END
85

Figure 3

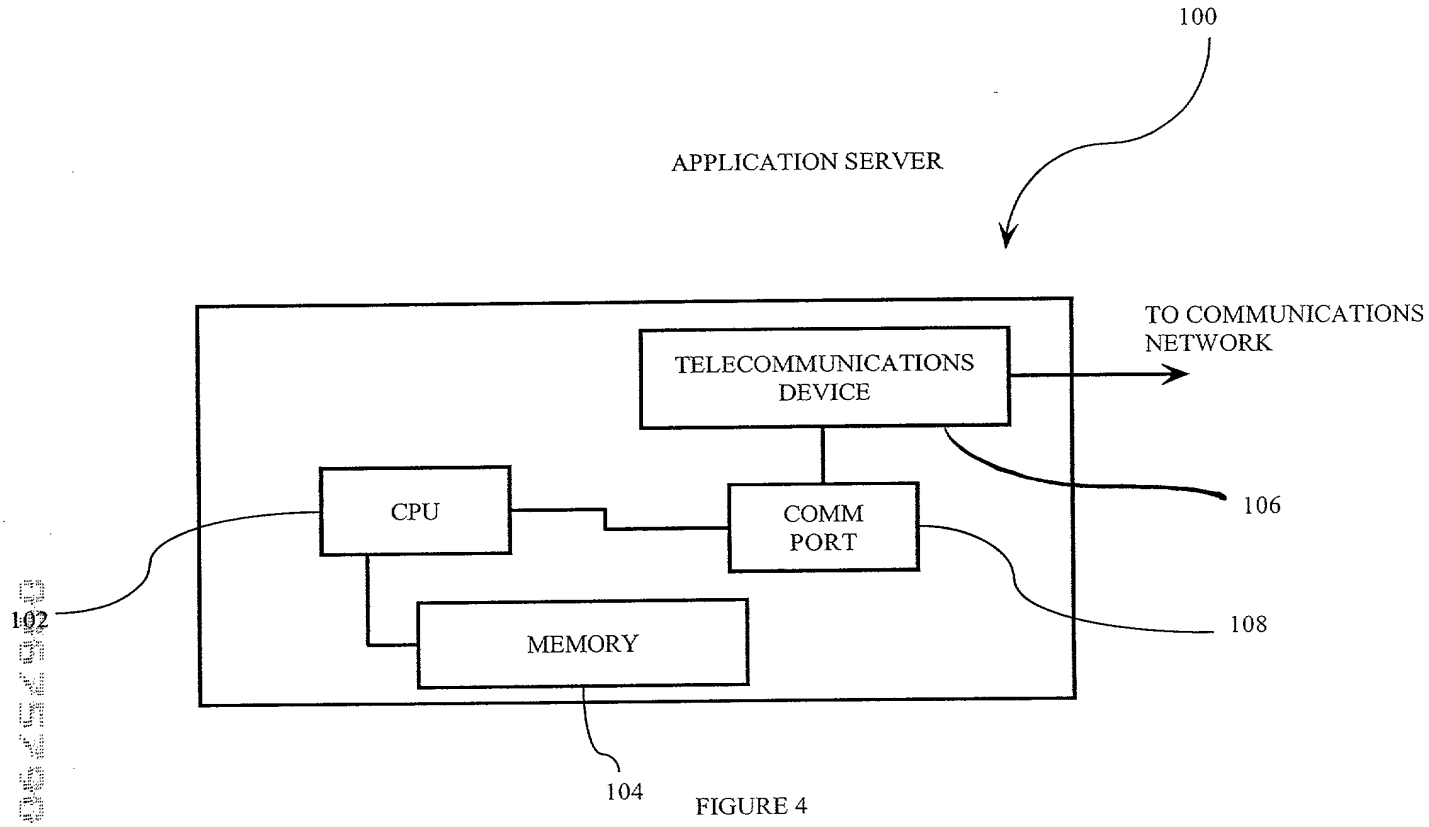


FIGURE 4

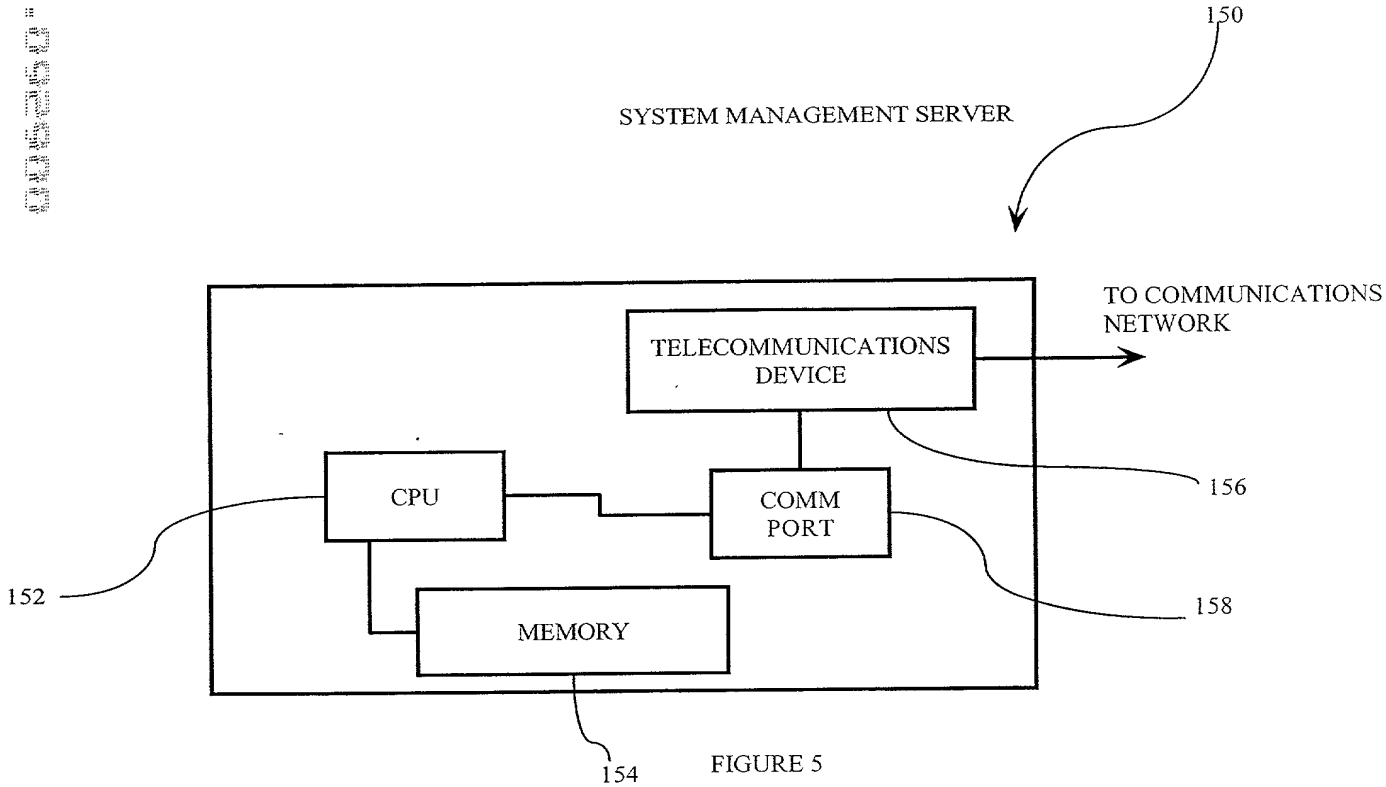


FIGURE 5

PATENT

Express Mail Label No. EL592133389US
Docket No. RSW9-2000-0119-US1

Declaration and Power of Attorney for Patent Application

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

SYSTEM AND METHOD FOR UPGRADING SOFTWARE IN A DISTRIBUTED COMPUTER SYSTEM

the specification of which:

☑ is attached hereto.

☐ was filed on _____ as Application Serial No. _____ and was

amended on _____.

I hereby state that I have reviewed and understand the contents of the above- identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):

Number	Country	Day/Month/Year	Priority Claimed
_____	_____	_____	_____

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

PATENT

Express Mail Label No. EL592133389US
Docket No. RSW9-2000-0119-US1

Prior U.S. Applications:

Serial No.	Filing Date	Status

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

A.B. Clay	Reg. No. 32,121
G.M. Doudnikoff	Reg. No. 32,847
E.H. Duffield	Reg. No. 25,970
J.W. Herndon	Reg. No. 27,901
J.S.Ray-Yarletts	Reg. No. 39,808
John T. Synnestvedt	Reg. No. 18,117
Charles H. Lindrooth	Reg. No. 20,659
Alexis Barron	Reg. No. 22,702
Peter J. Butch, III	Reg. No. 32,203
Joseph F. Posillico	Reg. No. 32,290
Mark D. Simpson	Reg. No. 32,942
Theodore Naccarella	Reg. No. 33,023
Patrick J. Kelly, Ph.D.	Reg. No. 34,638
Gary A. Hecht	Reg. No. 36,826
Stephen J. Driscoll	Reg. No. 37,564
Lisa B. Lane	Reg. No. 38,217
Joshua R. Slavitt	Reg. No. 40,816
John A. Chionchio	Reg. No. 40,954
Gregory S. Bernabeo	Reg. No. 44,032
Stephen J. Weed	Reg. No. 45,202
Brett T. Freeman	Reg. No. 46,709
Esther Chong	Reg. No. 40.953
Gene J. Yao	Reg. No. 47,193

PATENT

Express Mail Label No. EL592133389US
Docket No. RSW9-2000-0119-US1

Send all correspondence to:

Gregory S. Bernabeo, Esquire
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107-2950

Telephone: (215) 923-4466
Facsimile: (215) 923-2189

(1) Inventor: **Donald F. Ferguson**

Signature: Donald Ferguson Date: 9/29/00
Residence: **2440 Pine Tree Place, Yorktown, NY 10598**
Citizenship: **U.S.A.**
Post Office Address: **Same as Residence**

(2) Inventor: **Anand Byrappagari**

Signature: _____ Date: _____
Residence: **3102 Park Plaza Drive, Pittsburgh, PA 15229**
Citizenship: **India**
Post Office Address: **Same as Residence**

(3) Inventor: **Graeme Dixon**

Signature: _____ Date: _____
Residence: **2214 Bond Street, Pittsburgh, PA 15237**
Citizenship: **United Kingdom**
Post Office Address: **Same as Residence**

PATENT

Express Mail Label No. EL592133389US
Docket No. RSW9-2000-0119-US1

Declaration and Power of Attorney for Patent Application

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

SYSTEM AND METHOD FOR UPGRADING SOFTWARE IN A DISTRIBUTED COMPUTER SYSTEM

the specification of which:

☑ is attached hereto.

☐ was filed on _____ as Application Serial No. _____ and was

amended on _____.

I hereby state that I have reviewed and understand the contents of the above- identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s): Number	Country	Day/Month/Year	Priority Claimed

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Prior U.S. Applications: Serial No.	Filing Date	Status

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

A.B. Clay	Reg. No. 32,121
G.M. Doudnikoff	Reg. No. 32,847
E.H. Duffield	Reg. No. 25,970
J.W. Herndon	Reg. No. 27,901
J.S.Ray-Yarletts	Reg. No. 39,808
John T. Synnestvedt	Reg. No. 18,117
Charles H. Lindrooth	Reg. No. 20,659
Alexis Barron	Reg. No. 22,702
Peter J. Butch, III	Reg. No. 32,203
Joseph F. Posillico	Reg. No. 32,290
Mark D. Simpson	Reg. No. 32,942
Theodore Naccarella	Reg. No. 33,023
Patrick J. Kelly, Ph.D.	Reg. No. 34,638
Gary A. Hecht	Reg. No. 36,826
Stephen J. Driscoll	Reg. No. 37,564
Lisa B. Lane	Reg. No. 38,217
Joshua R. Slavitt	Reg. No. 40,816
John A. Chionchio	Reg. No. 40,954
Gregory S. Bernabeo	Reg. No. 44,032
Stephen J. Weed	Reg. No. 45,202
Brett T. Freeman	Reg. No. 46,709
Esther Chong	Reg. No. 40,953
Gene J. Yao	Reg. No. 47,193

Send all correspondence to:

Gregory S. Bernabeo, Esquire
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107-2950

Telephone: (215) 923-4466
Facsimile: (215) 923-2189

(1) Inventor: **Donald F. Ferguson**

Signature: _____ Date:_____
Residence: **2440 Pine Tree Place, Yorktown, NY 10598**
Citizenship: **U.S.A.**
Post Office Address: **Same as Residence**

(2) Inventor: **Anand Byrappagari**

Signature: _____ Date:_____
Residence: **3102 Park Plaza Drive, Pittsburgh, PA 15229**
Citizenship: **India**
Post Office Address: **Same as Residence**

(3) Inventor: **Graeme Dixon**

Signature: Graeme Dixon Date: 9/29/2000
Residence: **2214 Bond Street, Pittsburgh, PA 15237**
Citizenship: **United Kingdom**
Post Office Address: **Same as Residence**

PATENT

Express Mail Label No. EL592133389US
Docket No. RSW9-2000-0119-US1

Declaration and Power of Attorney for Patent Application

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

SYSTEM AND METHOD FOR UPGRADING SOFTWARE IN A DISTRIBUTED COMPUTER SYSTEM

the specification of which:

☑ is attached hereto.

☐ was filed on ____ __ as Application Serial No. _____ and was

amended on _____.

I hereby state that I have reviewed and understand the contents of the above- identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s): Number	Country	Day/Month/Year	Priority Claimed
_____	_____	_____	_____

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Prior U.S. Applications:

Serial No.	Filing Date	Status

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

A.B. Clay	Reg. No. 32,121
G.M. Doudnikoff	Reg. No. 32,847
E.H. Duffield	Reg. No. 25,970
J.W. Herndon	Reg. No. 27,901
J.S.Ray-Yarletts	Reg. No. 39,808
John T. Synnestvedt	Reg. No. 18,117
Charles H. Lindrooth	Reg. No. 20,659
Alexis Barron	Reg. No. 22,702
Peter J. Butch, III	Reg. No. 32,203
Joseph F. Posillico	Reg. No. 32,290
Mark D. Simpson	Reg. No. 32,942
Theodore Naccarella	Reg. No. 33,023
Patrick J. Kelly, Ph.D.	Reg. No. 34,638
Gary A. Hecht	Reg. No. 36,826
Stephen J. Driscoll	Reg. No. 37,564
Lisa B. Lane	Reg. No. 38,217
Joshua R. Slavitt	Reg. No. 40,816
John A. Chionchio	Reg. No. 40,954
Gregory S. Bernabeo	Reg. No. 44,032
Stephen J. Weed	Reg. No. 45,202
Brett T. Freeman	Reg. No. 46,709
Esther Chong	Reg. No. 40,953
Gene J. Yao	Reg. No. 47,193

Send all correspondence to:

Gregory S. Bernabeo, Esquire
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107-2950

Telephone: (215) 923-4466
Facsimile: (215) 923-2189

(1) Inventor: **Donald F. Ferguson**

Signature: _____ Date: _____
Residence: **2440 Pine Tree Place, Yorktown, NY 10598**
Citizenship: **U.S.A.**
Post Office Address: **Same as Residence**

(2) Inventor: **Anand Byrappagari**

Signature: B. Anand Byrappagari Date: 09/29/2000
Residence: **3102 Park Plaza Drive, Pittsburgh, PA 15229**
Citizenship: **India**
Post Office Address: **Same as Residence**

(3) Inventor: **Graeme Dixon**

Signature: _____ Date: _____
Residence: **2214 Bond Street, Pittsburgh, PA 15237**
Citizenship: **United Kingdom**
Post Office Address: **Same as Residence**